

Câu 1: Các nền tảng trên thiết bị di động

Android

Đặc điểm:

Mã nguồn mở: Cho phép các nhà phát triển tùy chỉnh và xây dựng các phiên bản Android riêng của họ.

Google Play Store: Nền tảng chính để phân phối và tải xuống các ứng dụng Android.

Đa dạng thiết bị: Hỗ trợ nhiều loại thiết bị từ nhiều nhà sản xuất khác nhau, dẫn đến sự phong phú về thiết kế và giá cả.

Tùy biến cao: Người dùng có thể tùy chỉnh giao diện, cài đặt các ứng dụng bên ngoài Google Play Store và thay đổi nhiều thiết lập hệ thống

Ưu điểm:

Sự đa dạng về thiết bị và giá cả, phù hợp với nhiều đối tượng người dùng.

Khả năng tùy biến và cài đặt ứng dụng linh hoạt.

Cộng đồng phát triển mạnh mẽ và phong phú

Nhược điểm:

Sự phân mảnh: Nhiều phiên bản Android cùng tồn tại làm khó khăn trong việc cập nhật và hỗ trợ.

Bảo mật: Do tính mở và khả năng cài đặt ứng dụng từ nguồn không chính thống, nguy cơ bảo mật cao hơn.

IOS

Đặc điểm:

Độc quyền: Chỉ có sẵn trên các thiết bị của Apple.

AppStore: Nền tảng duy nhất để phân phối và tải xuống các ứng dụng cho iOS.

Giao diện người dùng: Được thiết kế thân thiện và dễ sử dụng, với các ứng dụng tích hợp chất lượng cao.

Bảo mật: Tính năng bảo mật mạnh mẽ, bao gồm mã hóa dữ liệu, bảo mật sinh trắc học (Face ID, Touch ID).

Ưu điểm:

Tính ổn định và hiệu năng cao.

Bảo mật và bảo vệ quyền riêng tư mạnh mẽ.

Hệ sinh thái Apple đồng nhất và tương thích cao giữa các thiết bị.

Nhược điểm:

Độc quyền và giá cả cao: Giới hạn đối tượng người dùng.

Hạn chế tùy biến: Hạn chế trong việc tùy biến hệ điều hành và cài đặt ứng dụng từ bên ngoài AppStore.

Windows phone

Đặc điểm:

Giao diện Metro: Giao diện người dùng với các ô vuông động (Live Tiles), cung cấp thông tin cập nhật trực tiếp trên

màn hình chính.

Hệ sinh thái Microsoft: Tích hợp tốt với các dịch vụ của Microsoft như Office, OneDrive và Outlook.

Tính năng đồng bộ: Khả năng đồng bộ hóa mạnh mẽ với hệ điều hành Windows trên máy tính và các thiết bị khác của Microsoft.

Ưu điểm:

Giao diện người dùng trực quan và dễ sử dụng.

Tích hợp chặt chẽ với các dịch vụ của Microsoft.

Hiệu năng mượt mà trên các thiết bị cấu hình trung bình.

Nhược điểm:

Số lượng ứng dụng hạn chế so với Android và iOS.

Thiếu sự hỗ trợ và cập nhật từ Microsoft sau khi ngừng phát triển.

Ít sự lựa chọn về thiết bị

Câu 2:

Native Development (iOS và Android)

- Ngôn ngữ: Swift hoặc Objective-C cho iOS và Kotlin hoặc Java cho Android.

- Ưu điểm:

- Hiệu năng cao, tận dụng tối đa tài nguyên của hệ điều hành.
- Tích hợp sâu với các tính năng của thiết bị (GPS, camera, cảm biến, v.v.).
- Trải nghiệm người dùng tối ưu, giao diện tự nhiên và quen thuộc với người dùng.

- Nhược điểm:

- Phát triển riêng cho từng nền tảng, dẫn đến chi phí và thời gian cao.
- Khó bảo trì nếu muốn đồng bộ tính năng cho cả hai hệ điều hành.

Flutter

- Ngôn ngữ: Dart.

- Ưu điểm:

- Có thể phát triển ứng dụng đa nền tảng (iOS và Android) với một mã nguồn duy nhất.

- Hiệu năng gần như native nhờ rendering engine riêng.
- Khả năng tùy biến giao diện linh hoạt, hỗ trợ tạo UI đẹp mắt.

- Nhược điểm:

- Kích thước ứng dụng có thể lớn hơn.
- Các tính năng hoặc thư viện hỗ trợ có thể chưa phong phú bằng native

React Native

- Ngôn ngữ: JavaScript.

- Ưu điểm:

- Một mã nguồn có thể dùng cho cả iOS và Android.
- Hỗ trợ mạnh từ cộng đồng, tích hợp dễ dàng với hệ sinh thái JavaScript.
- Nhiều thư viện và công cụ hỗ trợ phát triển UI.

- Nhược điểm:
 - Hiệu năng có thể kém hơn so với native trong các ứng dụng phức tạp.
 - Gặp khó khăn khi xử lý các tính năng phức tạp của thiết bị.

Xamarin

- Ngôn ngữ: C# và .NET.
- Ưu điểm:
 - Một mã nguồn có thể chạy trên nhiều nền tảng, tận dụng tối đa code tái sử dụng.
 - Tích hợp tốt với Microsoft và Azure.
 - Cho phép truy cập trực tiếp các API native của iOS và Android.
- Nhược điểm:
 - Hiệu năng có thể chậm hơn so với native.
 - Kích thước ứng dụng thường lớn hơn, nhất là với ứng dụng đơn giản.

Câu 3

Điều gì làm cho Flutter trở thành một lựa chọn phổ biến cho việc phát triển ứng dụng đa nền tảng là vì

Hiệu suất: Chọn Flutter nếu yêu cầu hiệu suất cao.

Giao diện người dùng: Chọn Flutter nếu yêu cầu giao diện người dùng đẹp và nhất quán trên nhiều nền tảng

Cộng đồng: Flutter cộng đồng lớn và tích cực.

Mã nguồn mở: là mã nguồn mở.

Tính linh hoạt: Flutter dễ dàng tích hợp với các thư viện bên ngoài và có tính tùy biến cao.

Tiêu chí	Flutter	React Native	Xamarin
Ngôn ngữ lập trình	Dart	Javascript	C#
Hiệu suất	Biên dịch trực tiếp thành mã máy, hiệu suất cao	Sử dụng các thành phần UI gốc, hiệu suất gần gốc	Truy cập đầy đủ API gốc, hiệu suất cao
Hệ thống Widget	Hệ thống widget phong phú, tùy biến cao	Component-based, sử dụng các component UI gốc	Xamarin.Forms cho phép chia sẻ mã nguồn UI
Hot Reload	Có	Có	Có
Ưu điểm	Tốc độ phát triển nhanh, giao diện đẹp, cộng đồng hỗ trợ mạnh mẽ	Sử dụng JavaScript phổ biến, tích hợp tốt với hệ sinh thái React, hiệu suất cao	Chia sẻ mã nguồn lớn, sử dụng C#, tích hợp tốt với dịch vụ Microsoft
Nhược điểm	Dart chưa phổ biến rộng rãi, kích thước ứng dụng lớn	Yêu cầu kiến thức lập trình gốc, cộng đồng không đồng đều	Kích thước ứng dụng lớn, cộng đồng nhỏ hơn

Bảng so sánh 3 công nghệ Flutter, React Native và Xamarin

Câu 4

Ngôn ngữ	Lý do chính được chọn
Java	Ngôn ngữ chính thức truyền thống, ổn định, nhiều tài liệu Dự án lớn cần ổn định, dễ bảo trì
Kotlin	Ngắn gọn, an toàn, hiện đại Phát triển ứng dụng mới, dễ đọc và bảo trì
C++	Hiệu năng cao, kiểm soát tài nguyên
Dart	Đa nền tảng, phát triển nhanh với Flutter

Câu 5

Objective-C: Đây là ngôn ngữ lập trình truyền thống dùng cho phát triển ứng dụng iOS trước khi có Swift. Objective-C vẫn được sử dụng trong các dự án lớn và những dự án duy trì.

Swift: Được phát triển bởi Apple, Swift hiện nay là ngôn ngữ chính để phát triển ứng dụng trên iOS nhờ vào cú pháp dễ hiểu, hiệu suất cao và hỗ trợ tốt từ Apple.

C/C++: Đôi khi được sử dụng cho các phần mềm cần hiệu suất cao hoặc yêu cầu xử lý gần hệ điều hành, mặc dù ít phổ biến hơn.

Câu 6

Hệ sinh thái ứng dụng yếu: Windows Phone thiếu hụt các ứng dụng phổ biến so với iOS và Android, dẫn đến trải nghiệm người dùng kém. Các nhà phát triển thường ưu tiên phát triển ứng dụng cho iOS và Android trước do hai hệ điều hành này có lượng người dùng lớn hơn.

Chậm chạp trong việc cập nhật và cải tiến: Microsoft không thể theo kịp tốc độ cải tiến của Apple và Google. Các bản cập nhật của Windows Phone thường không mang lại nhiều tính năng mới và không đủ để thu hút người dùng.

Thiếu sự hỗ trợ từ các đối tác phần cứng: Không có nhiều hãng sản xuất điện thoại hỗ trợ mạnh mẽ cho Windows Phone. Điều này khiến hệ điều hành không được đa dạng về mẫu mã thiết bị, làm giảm sức hấp dẫn đối với người tiêu dùng.

Chiến lược tiếp thị và phát triển kém hiệu quả: Microsoft gặp khó khăn trong việc xây dựng thương hiệu và thuyết phục người dùng chuyển sang Windows Phone. Cách tiếp cận thị trường và quảng bá của họ không đủ mạnh mẽ so với Apple và Google.

Thị phần đã bị chiếm lĩnh bởi Android và iOS: Khi Windows Phone ra mắt, thị trường smartphone đã gần như bị thống trị bởi Android và iOS. Sự gia nhập muộn này khiến Microsoft khó có thể thu hút được người dùng mới.

Câu 7

Ngôn ngữ

HTML, CSS, và JavaScript: Đây là các ngôn ngữ cơ bản nhất cho phát triển ứng dụng web trên mọi thiết bị, bao gồm thiết bị di động.

TypeScript: Là một phiên bản mở rộng của JavaScript, TypeScript giúp mã nguồn dễ duy trì và quản lý hơn, đặc biệt khi dự án lớn.

Dart: Được phát triển bởi Google và thường được sử dụng với Flutter để tạo ứng dụng web và di động.

Công cụ

Visual Studio Code: Một trình soạn thảo mã nguồn mở và miễn phí, rất phổ biến cho việc phát triển web và hỗ trợ nhiều plugin.

Xcode và Android Studio: Cần thiết nếu muốn triển khai ứng dụng web trên các thiết bị iOS và Android.

Câu 8

Tiền lương khoảng 10 triệu